

## CLAIMS

What is claimed is:

1. A method comprising:  
analyzing a stride profile, and  
inserting a prefetch instruction immediately before a load instruction using  
stride profiling information.
2. The method of claim 1, further comprising the steps of identifying  
candidate loads, grouping candidate loads and selected profiled loads, inserting  
profiling instructions, and collecting a stride profile analysis.
3. The method of claim 2, further comprising the step of collecting a top N  
most frequently occurring stride value and frequency to provide a top stride  
profile.
4. The method of claim 2, further comprising the step of profiling the  
difference of successive strides to collect the top M most frequently occurred  
differences and their frequencies to provide a top differential profile to distinguish  
phased stride sequences from alternated stride sequences.
5. The method of claim 1, further comprising the step of analyzing range of  
cache area accessed by a load in a loop, and inserting a prefetch instruction at  
the additive combination of a load address P and a determined compile time  
constant.
6. The method of claim 5, further comprising the step of determining a  
prefetching distance from at least one of a cache profile and a compiler analysis.

7. The method of claim 1, further comprising determining a cache profile to assist in determining appropriate insertion of a prefetch instruction.

8. An article comprising a computer-readable medium which stores computer-executable instructions, the instructions causing a computer to:  
analyze a stride profile for code;  
insert a prefetch instruction immediately before a load instruction using stride profiling information.

9. The article comprising a computer-readable medium which stores computer-executable instructions of claim 8, wherein the instructions further cause a computer to identify candidate loads, group candidate loads and selected profiled loads, insert profiling instructions, and collect a stride profile analysis.

10. The article comprising a computer-readable medium which stores computer-executable instructions of claim 9, wherein the instructions further cause a computer to collect a top N most frequently occurring stride value and frequency to provide a top stride profile.

11. The article comprising a computer-readable medium which stores computer-executable instructions of claim 8, wherein the instructions further cause a computer to profile the difference of successive strides to collect the top M most frequently occurred differences and their frequencies to provide a top differential profile to distinguish phased stride sequences from alternated stride sequences.

12. The article comprising a computer-readable medium which stores computer-executable instructions of claim 9, wherein the instructions further cause analyzing range of cache area accessed by a load in a loop iteration, and

insertion of a prefetch instruction at the additive combination of a load address P and a determined compile time constant.

13. The article comprising a computer-readable medium which stores computer-executable instructions of claim 9, wherein the instructions further cause determination of a prefetching distance from at least one of a cache profile and a compiler analysis.

14. The article comprising a computer-readable medium which stores computer-executable instructions of claim 9, wherein the instructions further cause determination of a cache profile to assist in determining appropriate insertion of a prefetch instruction.

15. A system for optimizing software comprising:  
an analyzing module for determining a stride profile; and  
an optimizing module for inserting a prefetch instruction immediately before a load instruction using stride profile.

16. The system of claim 15 for optimizing software further comprising:  
a stride profiling module that identifies candidate loads, groups candidate loads and selected profiled loads, inserts profiling instructions, and executes and instrumented program.

17. The system of claim 16 for optimizing software wherein the stride profiling module collects a top N most frequently occurring stride value and frequency to provide a top stride profile.

18. The system of claim 16 for optimizing software wherein the stride profiling module profiles the difference of successive strides to collect the top M most frequently occurred differences and their frequencies to provide a top differential profile to distinguish phased stride sequences from alternated stride sequences.

19. The system of claim 15 for optimizing software wherein the optimizing module analyzes a range of cache area accessed by a load in a loop iteration, and inserts a prefetch instruction at the additive combination of a load address P and a determined compile time constant.

20. The system of claim 19 for optimizing software wherein the optimizing module determines a prefetching distance from at least one of a cache profile and a compiler analysis.

21. The system of claim 19 for optimizing software wherein the analyzing module determines a cache profile to provide information to the optimizing module.